

Cont 83  
the entry and exit handlers being programmed to maintain an association between one of the threads and an extended context of the thread through a context change induced by the operating system, the extended context including resources of the computer associated with the thread beyond those resources whose association with the thread is maintained by the operating system.

---

75. (amended) The method of claim 56:

C4  
wherein the interrupted thread at the point of interruption executes in one instruction set architecture and the operating system is coded primarily in a different instruction set architecture; and

further comprising the step of setting of a register to a value that specifies actions to be taken by the entry handler or exit handler to convert operands from one form to another to conform to a data storage convention of the operating system instruction set architecture.

---

#### REMARKS

This paper responds to the Office Action of April 29, 2003. The shortened statutory period runs through July 29, 2003, and thus this paper is timely.

Applicant respectfully requests reconsideration of the application. Claims 1-83 are now pending, a total of 83 claims. Claims 1, 5, 33, 56 and 79 are independent. Of the independent claims, the Office Action discusses claims 79 in the context of § 102(e) and U.S. Patent 5,652,869 (Herdeg), and claims 1, 5, 33 and 56 in the context of § 102(e) and U.S. Patent No 6,081,665 (Nilsen). The remaining claims are discussed in connection with §§ 102(e) and 103(a) and a variety of other references.

Applicant thanks the Examiner for a telephone interview of July 2, 2003. Applicant and Examiner discussed an overview of the technology, and several particular claim limitations, as discussed below.

The Office Action of April 29, 2003 and Information Disclosure Statements filed December 31, 2002, April 16, 2003 and May 2, 2003 crossed in the mail. Applicant requests an initialed copy of these IDS's with the next paper.

Applicant filed a Preliminary Amendment on January 28, 1999 that made minor amendments to the specification. The Office Action of April 29, 2003 does not mention that that Preliminary Amendment was entered. Kindly cancel that Preliminary Amendment, and do not enter it. The formal drawings submitted herewith assume that the Preliminary Amendment will not be entered.

**I. § 112 issues**

Paragraph 7 of the Action questions the term “exit exception” in claim 1. Claim 1 is now amended to recite “resumption exception,” and that term is defined earlier in the claim:

... establishing a resumption exception to be raised on each resumption from the operating system complementary to one of the specified entries, the resumption exception having an associated exit handler, ...

It is believed that this resolves any question.

**II. Claim 5**

The Office Action discusses claim 5 in paragraph 13 in connection with the Nilsen '665 patent. Claim 5 recites as follows:

5. A method, comprising:

scheduling concurrent threads of control by a pre-existing thread scheduler of a computer, each thread having an associated context, an association between a thread and a set of computer resources of the associated context being maintained by the thread scheduler; and

without modifying the thread scheduler, maintaining an association between one of the threads and an extended context of the thread through a context change induced by the thread scheduler, the extended context including resources of the computer associated with the thread beyond those resources whose association with the thread is maintained by the thread scheduler.

In the interview, it was agreed that col. 25, lines 40-67 and col. 37, lines 60-67 of the Nilsen '665 patent do not discuss any feature analogous to the “extended context” recited in claim 5. For that reason, Applicant submits that claim 5 is patentable over the Nilsen '667 patent.

Claim 1 recites similar limitations and is patentable for similar reasons. Claims 2-4, 6-32 and 82-83 are dependent on these independent claims, and patentable therewith, and recite further patentably-distinct limitations.

### **III. Claim 33**

The Office Action discusses claim 33 in paragraph 13 in connection with the Nilsen '665 patent. Claim 33 recites as follows:

33. A method, comprising:
- establishing an entry exception to be raised on each entry to a computer operating system at a specified entry point or on a specified condition;
  - establishing a resumption exception to be raised on each resumption from the operating system complementary to one of the specified entries;
  - on detecting a specified entry to the operating system from an interrupted process of the computer, raising and servicing the entry exception, and then entering the operating system to perform a service associated with the original operating system entry; and
  - on detecting a complementary resumption, raising and servicing the resumption exception, and returning control to the interrupted process.

In the interview, it was agreed that col. 25, lines 40-67, col. 26, lines 6-15, col. 33, lines 40-67 and col. 37, lines 60-67 of the Nilsen '665 patent do not discuss any feature analogous to the "entry exception" and "exit exception" recited in claim 33. Further, Nilsen '667 makes only minor mention of the "operating system," and never in a context that relates to the "operating system" limitations of the claim. For these reasons, Applicant submits that claim 33 is patentable over the Nilsen '667 patent.

Claim 1 recites similar limitations and is patentable for similar reasons. Claims 2-4 and 34-55 are dependent on these independent claims, and patentable therewith, and recite further patentably-distinct limitations.

### **IV. Claim 56**

Paragraph 13 of the Office Action discusses claim 56 in connection with the Nilsen '665 patent. Claim 56 recites as follows:

56. A method, comprising:
- without modifying a pre-existing operating system of the computer, establishing an entry handler for execution at a specified entry point or on a specified entry condition to the operating system, the entry handler programmed to save a context of an interrupted thread and modify the thread context before delivering the modified context to the operating system;
  - without modifying the operating system, establishing an exit handler for execution on resumption from the operating system following an entry through the entry handler, the exit handler programmed to restore the context saved by a corresponding execution of the entry handler.

In the interview, it was agreed that col. 25, lines 40-67, col. 26, lines 6-15, col. 33, lines 40-67 and col. 37, lines 60-67 of the Nilsen '665 patent do not discuss any feature analogous to the "entry handler" and "exit handler" recited in claim 33. Further, Nilsen '667 makes only minor mention of the "operating system," and never in a context that relates to the "operating system" limitations of the claim. For these reasons, Applicant submits that claim 56 is patentable over the Nilsen '667 patent.

Claim 1 recites similar limitations and is patentable for similar reasons. Claims 2-4 and 57-78 are dependent on these independent claims, and patentable therewith, and recite further patentably-distinct limitations.

#### **V. Claim 79**

Paragraph 10 of the Office Action discusses claim 79 in connection with the Herdeg '869 patent. Claim 79 recites as follows:

79. A method, comprising:

during invocation of a service routine of a computer, passing a linkage return address to the service routine at which to resume execution on completion of the service, the linkage return address being deliberately chosen so that an attempt to execute an instruction from the linkage return address on return from the service routine will raise a program execution exception;

on return from the service routine, attempting to execute the instruction at the linkage return address and raising the chosen exception; and

after servicing the exception, returning control to a caller of the service routine.

In the interview, it was agreed that the "linker program" of Herdeg '869 is unrelated to the "linkage return address" of the claim, and that col. 13, lines 19-24 do not discuss any feature analogous to the deliberate "program execution exception" of claim 79. For these reasons, Applicant submits that claim 79 is patentable over the Herdeg '869 patent.

Claim 1 recites similar limitations and is patentable for similar reasons. Claims 2-4 and 80-81 are dependent on these independent claims, and patentable therewith, and recite further patentably-distinct limitations.

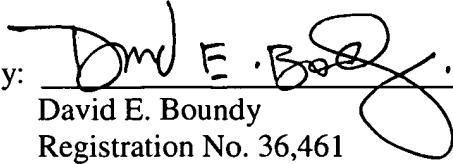
In view of the amendments and remarks, Applicant respectfully submits that the claims are in condition for allowance. Applicant requests that the application be passed to issue in due course. The Examiner is urged to telephone Applicant's undersigned counsel at the number

noted below if it will advance the prosecution of this application, or with any suggestion to resolve any condition that would impede allowance. In the event that any extension of time is required, Applicant petitions for that extension of time required to make this response timely. Kindly charge any additional fee, or credit any surplus, to 50-0675, Order No. 5231.05-4013.

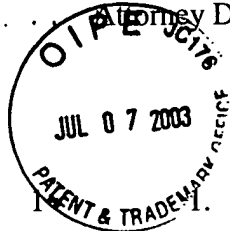
Respectfully submitted,

SCHULTE ROTH & ZABEL, LLP

Dated: July 3, 2003

By:   
David E. Boundy  
Registration No. 36,461

Mailing Address:  
SCHULTE ROTH & ZABEL, LLP  
919 Third Avenue  
New York, New York 10022  
(212) 756-2000  
(212) 593-5955 Telecopier

**REWRITTEN CLAIMS MARKED UP TO SHOW CHANGES**

1. A method, comprising:

2 without modifying a pre-existing operating system of the computer, establishing an entry  
3 exception to be raised on each entry to the operating system at a specified entry point or on a  
4 specified condition, the entry exception having an associated entry handler, the entry handler  
5 programmed to save a context of an interrupted thread and modify the thread context before  
6 delivering the modified context to the operating system;

7 without modifying the operating system, establishing a resumption exception to be raised  
8 on each resumption from the operating system complementary to one of the specified entries, the  
9 resumption exception having an associated exit handler, the exit handler programmed to restore  
10 the context saved by a corresponding execution of the entry handler.

11 scheduling concurrent threads of control by the operating system, each thread having an  
12 associated context, the association between a thread and a set of computer resources of the  
13 associated context being maintained by the operating system;

14 on detecting a specified entry to the operating system from an interrupted thread of the  
15 computer, raising and servicing the entry exception; and

16 on detecting a complementary resumption, raising and servicing the resumption  
17 exception, and returning control to the interrupted thread;

18 the entry exception, resumption [exit] exception, entry handler, and exit handler being  
19 cooperatively designed to maintain an association between [a] one of the threads and an extended  
20 context of the thread through a context change induced by the operating system, the extended  
21 context including resources of the computer associated with the thread beyond those resources  
22 whose association with the thread is maintained by the operating system.

1 5. A method, comprising:

2 scheduling concurrent threads of control by a pre-existing thread scheduler of a  
3 computer, each thread having an associated context, an association between a thread and a set of  
4 computer resources of the associated context being maintained by the thread scheduler; and

5 without modifying the thread scheduler, maintaining an association between [a] one of  
6 the threads and an extended context of the thread through a context change induced by the thread

7 scheduler, the extended context including resources of the computer associated with the thread  
8 beyond those resources whose association with the thread is maintained by the thread scheduler.

57. The method of claim 56, further comprising:

scheduling concurrent threads of control by the operating system, each thread having an associated context, an association between a thread and a set of computer resources of the associated context being maintained by the operating system; and

the entry and exit handlers being programmed to maintain an association between [a] one of the threads and an extended context of the thread through a context change induced by the operating system, the extended context including resources of the computer associated with the thread beyond those resources whose association with the thread is maintained by the operating system.

75. The method of claim 56; [,]

wherein the interrupted thread at the point of interruption executes in one instruction set architecture and the operating system is coded primarily in a different instruction set architecture; and

further comprising the step of setting of a register to a value that specifies actions to be taken by the entry handler or exit handler to convert operands from one form to another to conform to a data storage convention of the operating system instruction set architecture [execution mode].